SCRUM

Overview

v1.1

Release PLAN

- **Incrément 1 – Gestion de projet agile**
  - **Sprint 1** Pourquoi Agile ?
  - **Sprint 2** Manifeste Agile, historique et points communs des méthodes
- **Incrément 2 – le Framework SCRUM**
  - **Sprint 1** Introduction
  - **Sprint 2** Equipe SCRUM
  - **Sprint 3** Cérémonies SCRUM
  - **Sprint 4** Artéfacts SCRUM
- **Incrément 3 – Et après**
  - **Sprint 1** Appliquer Scrum dans la réalité
  - **Sprint 2** Ressources & Certifications

1 heure 30

10/01/2016

G. Bladier

Commençons avec l'incrément 1

Incrément 1 – Gestion de projet agile

- **Sprint 1** Pourquoi Agile ?
- **Sprint 2** Manifeste Agile, historique et points communs des méthodes

- **Incrément 2 – le Framework SCRUM**
  - **Sprint 1** Introduction
  - **Sprint 2** Equipe SCRUM
  - **Sprint 3** Cérémonies SCRUM
  - **Sprint 4** Artéfacts SCRUM

- **Incrément 3 – Et après**
  - **Sprint 1** Appliquer Scrum dans la réalité
  - **Sprint 2** Ressources & Certifications

Pourquoi Agile ?

Contexte : Parfois, *mais plus spécifiquement dans les projets IT*, il est impossible de collecter l'ensemble des exigences produit ou projet en amont à cause des **incertitudes** et des **inconnues**.

Problématique : Les méthodes de gestion de projet traditionnelles n'embrassent pas le changement et échoue à délivrer un produit pleinement conforme à l'attendu.

Le besoin : Pour gérer certains de ces projets (plus) efficacement le besoin d'une nouvelle méthode de gestion de projet qui répondrait au cahier des charges suivant :

- Proposer plus de flexibilité et d'adaptabilité au changement qui apparait tout au long du projet.
- Garder l'équipe projet productive

La réponse : le besoin a déclenché un véritable « choc » culturel dans la gestion de projet.

On parle de culture agile, d'un nouvel état d'esprit ou encore du paradigme agile. Le terme méthode est trop restrictif :

- Dans les années 90 de nombreux Framework ont été créés, mais la 1ère technique remonte aux années 1950s,
- **Scrum** est probablement l'ambassadeur le plus connu et le plus utilisé dans le monde.

Le manifeste Agile 1/2 – Les valeurs piliers

En 2001, le manifeste a été publié pa un groupe de spécialistes en développement informatique et en gestion de projet.

→ Depuis, il a été adopté et considéré comme l'essence fondamentale des méthodes Agiles.

→ Scrum est une manière d'implémenter ce manifeste, il y en a plein d'autres (AUP, XP, Lean, Kanban, FDD, Crystal Clear)

Définition du manifeste agile :

We are uncovering better ways of developing ~~software~~ product by doing it and helping others do it.

A l'issue de la réunion ils ont mis en avant les valeurs suivantes :

CUSTOMER COLLABORATION	**INDIVIDUALS AND INTERACTIONS**	**RESPONDING TO CHANGE**	**WORKING SOFTWARE**
Over Contract Negociation	Over Processes and Tools	Over Following a Plan	Over Full Documentation

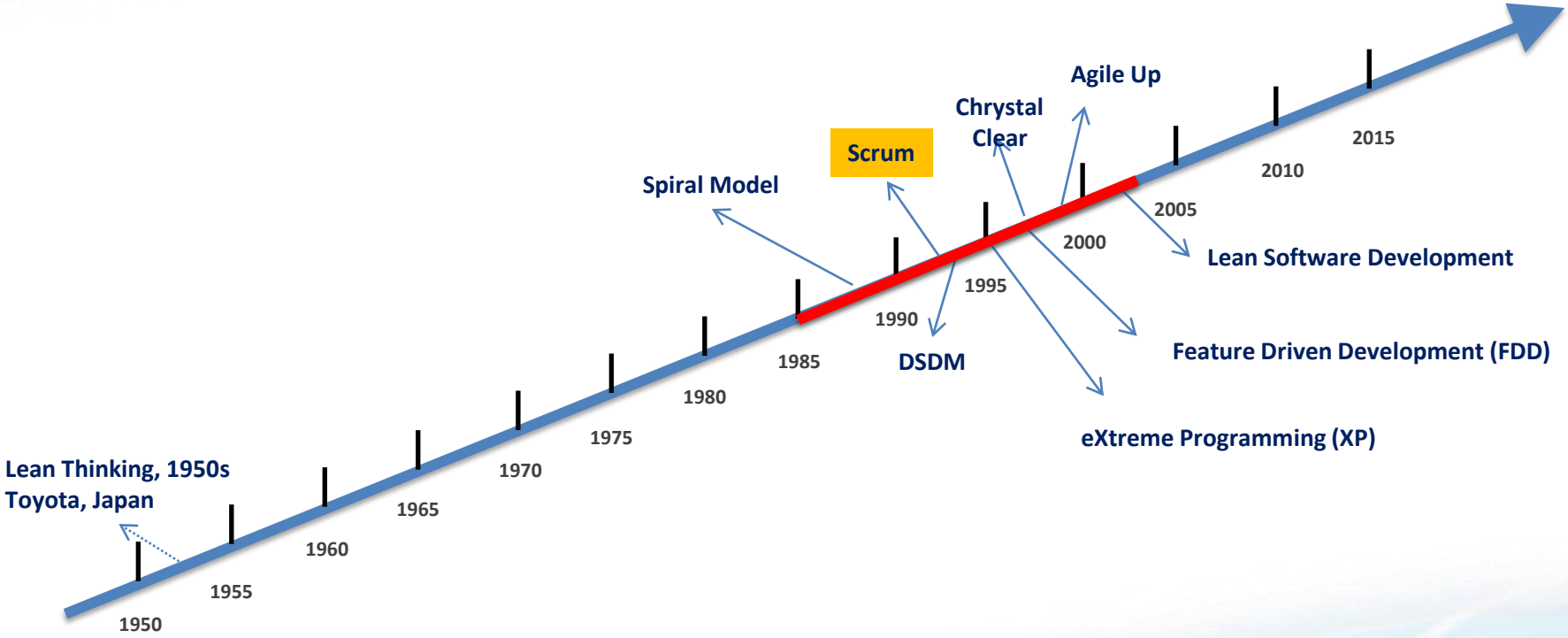
Bien qu'il y ait de la valeur dans l'item du bas (en gris)
Le manifeste privilégie les items du haut (en bleu)

Le manifeste Agile 2/2 – Les 12 principes

1. La priorité est de **satisfaire le client** en **livrant rapidement** et **régulièrement** des fonctionnalités à haute valeur ajoutée.
2. Il faut avoir une **attitude positive** envers les **changements**, même lorsqu'ils arrivent tardivement. C'est un avantage compétitif au client.
3. Il faut livrer régulièrement un logiciel **opérationnel** (utilisable en production) avec des **cycles courts** (4 semaines max).
4. Les ***utilisateurs*** (ou leurs représentants) et les ***développeurs*** **doivent travailler ensemble** quotidiennement et tout au long du projet.
5. Il convient de **s'appuyer** sur des **personnes motivées**, leur fournir des environnements adaptés, les soutenir et avoir confiance dans leur capacité à atteindre les objectifs fixés.
6. Le **dialogue en face à face** est la méthode la plus **simple** et la plus **efficace** pour communiquer.
7. **L'aspect opérationnel** d'un produit est sa principale mesure d'avancement.
8. Le **rythme de développement** doit être **soutenable** pour l'équipe et **constant** → pas de rushs permanents ou de montagnes russes
9. La recherche de l'excellence et de la performance conceptuelle et technique renforce l'agilité d'un produit.
10. Simplifier le travail en réduisant les efforts et tâches inutiles et redondants est essentiel.
11. Les **meilleures solutions logicielles** émergent **d'équipes auto-organisées** aussi bien au niveau de la clarté des spécifications, que de la conception et des architectures déployées.
12. L'équipe doit **adopter une démarche d'amélioration continue**, s'auditer pour s'améliorer et devenir plus efficace.

http://www.agilemanifesto.org

Les méthodes agiles : l'historique



Les méthodes agiles : les points communs

Toutes les méthodes agiles se ressemblent, et convergent vers le même mode opératoire :

1. Un responsable fonctionnel définit et priorise la production des livrables
2. Le projet est structuré en incréments courts de quelques semaines, dimensionnés en fonction du projet
3. Une réunion initiale sert à planifier l'incrément de manière détaillée (tâches)
4. Tous les jours, une réunion très courte permet à l'équipe de partager l'avancement, mettre en évidence les problèmes rencontrés.
5. L'équipe pilote elle-même ses activités, sa performance et la qualité des livrables
6. La progression quotidienne est affichée de manière transparent (Kanban, graphes de progression, etc.) et est mis à jour en temps réel par les membres de l'équipe.
7. Chaque incrément a pour objectif une livraison complète, développée, approuvée et testée.
8. Une réunion finale présente l'application et est suivie d'une rétrospective technique du processus de développement.
9. Le responsable fonctionnel valide le travail de l'équipe et ajuste les besoins entre chaque incrément.

Increment 2 - Sprint 1

- **Incrément 1 – Gestion de projet agile**
  - Sprint 1 Pourquoi Agile ?
  - Sprint 2 Manifeste Agile, historique et points communs des méthodes

- **Incrément 2 – le Framework SCRUM**
  - **Sprint 1** Introduction
  - Sprint 2 Equipe SCRUM
  - Sprint 3 Cérémonies SCRUM
  - Sprint 4 Artéfacts SCRUM

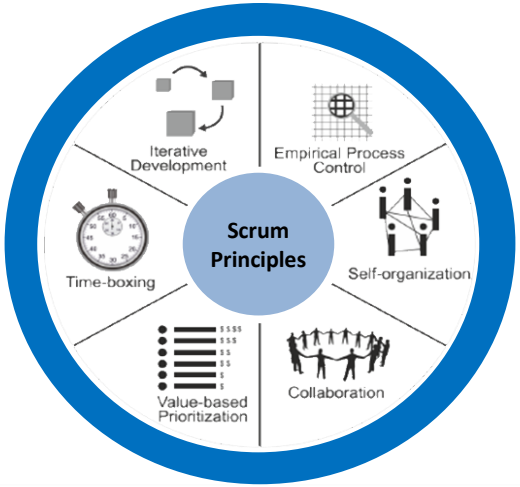
- **Incrément 3 – Et après**
  - Sprint 1 Appliquer Scrum dans la réalité
  - Sprint 2 Ressources & Certifications

The SCRUM Framework – Introduction 1/3

Scrum est un Framework pour développer et maintenir des produits complexes.

Autrement dit, Scrum est un Framework de gestion de projet "simple" *(ensemble de principes et processus)*
qui permet de délivrer des produits dans une approche itérative et incrémentale

6 Principes clés de Scrum
(Non officiel : Source Scrum Body of Knowledge)



1. **Empirical Process Control** – Scrum est construite autour de 3 idées fondamentales qui sont la *transparence*, l'*inspection*, et *adaptabilité*. (décrit slide suivante)
2. **Self-organization** – Les employés d'aujourd'hui sont plus à même de délivrer un produit de haute valeur ajoutée lorsqu'ils s'auto-organisent. Scrum y voit les avantages suivants : une meilleure adhésion au projet, un partage du mérite, une capacité d'innovation et un environnement plus propice à la croissance.
3. **Collaboration** – Ce principe s'appuie sur les 3 dimensions fondamentales du travail d'équipe : *conscience d'équipe*, l'*articulation*, et l'*appropriation*.
4. **Value-based Prioritization** – Scrum se **focalise** sur la *valeur maximale du produit*, et ce dès le démarrage jusqu'à la fin du projet.
5. **Time-boxing** – Le *temps est considéré comme une contrainte* dans Scrum. Il doit être utilisé de manière efficace pour gérer la planification et l'exécution du projet. Scrum limite dans le temps les *Sprint Daily Meetings*, *Sprint Planning Meetings*, *Sprint Review Meetings* and *Sprint Retrospective Meetings*.
6. **Iterative Development** – Scrum s'appuie sur un **développement itératif** c'est-à-dire que l'on va répéter pour chaque incrément un ensemble d'étapes clairement définies et adaptées au projet (conception, développement, tests, intégration, qualification par exemple)

The SCRUM Framework – Introduction 2/3

Le 1er pillier, ***1 - empirical process control theory***, est aussi appelé ***empirisme***.

→ Empirisme vérifie que tout savoir provient du fruit des expériences passées (On prend des décisions à partir de ce que l'on connait).

→ Scrum s'articule autour d'une démarche itérative et incrémentale afin d'optimiser la probabilité de réussite et de réduire les risques.

→ Les 3 piliers de l'empirisme sont : l'*inspection*, l'*adaptabilité* et la *transparence*

En substance :

- **Inspecter et s'adapter :** *Scrum va promouvoir l'inspection au sein des artéfacts et cérémonials*
  - Révision des plans et des hypothèses régulièrement
  - Optimisation de la valeur produite par l'équipe
  - Optimisation des processus

  → Scrum formalise 4 **événements** (rituels) pour l'inspection et adaptabilité :

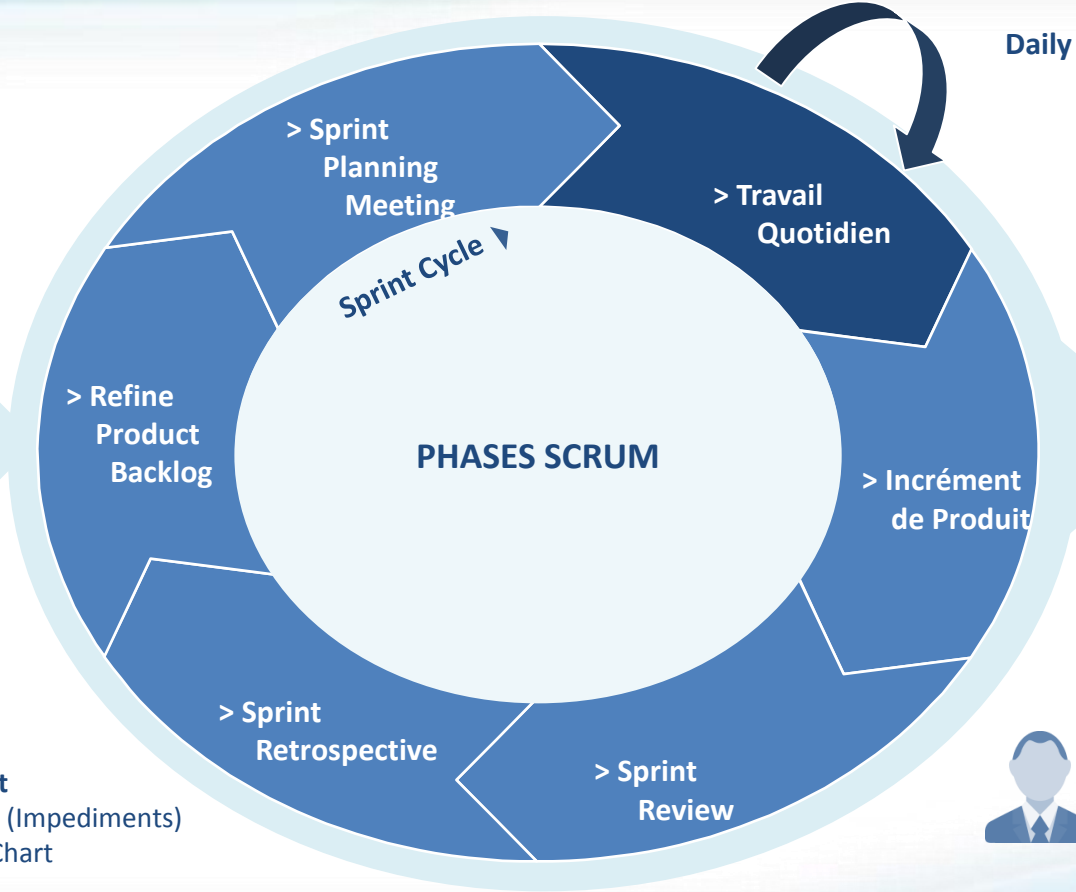
  Sprint Planning / *Daily Scrum* / *Sprint Review* / *Sprint Retrospective*

- **Etre transparent :** *Les tenants et aboutissants du projet doivent être visibles (accessibles) à ceux qui en sont responsables / impactés*
  - Définition d'un langage commun et partagé (ex: les "Definition of Ready" & " Definition of Done")
  - Rendre le travail effectué visible à l'équipe projet et aux parties prenantes
  - Piloter la "communication" au sein du projet et en dehors

  → l'information est un élément clé de la réussite d'un projet

The SCRUM Framework – Introduction 3/3

- Business Case & Vision du projet
- Accords contractuels
- Financement du projet
- Release Plan
- Product Backlog Initial
- Identification des PP & buy-in
- Acquisition de l'équipe



Releases

Artéfacts :
- **Product Backlog**
- **Sprint Backlog**
- **Product Increment**
- Liste des obstacles (Impediments)
- Sprint Burndown Chart
- …

ROLES :
- **Product Owner**
- **Development Team**
- **Scrum Master**
- Client
- Utilisateur
- Management
- Autres PP

Increment 2 - Sprint 2

- **Incrément 1 – Gestion de projet agile**
  - Sprint 1 Pourquoi Agile ?
  - Sprint 2 Manifeste Agile, historique et points communs des méthodes

- **Incrément 2 – le Framework SCRUM**
  - Sprint 1 Introduction
  - Sprint 2 Equipe SCRUM
  - Sprint 3 Cérémonies SCRUM
  - Sprint 4 Artéfacts SCRUM

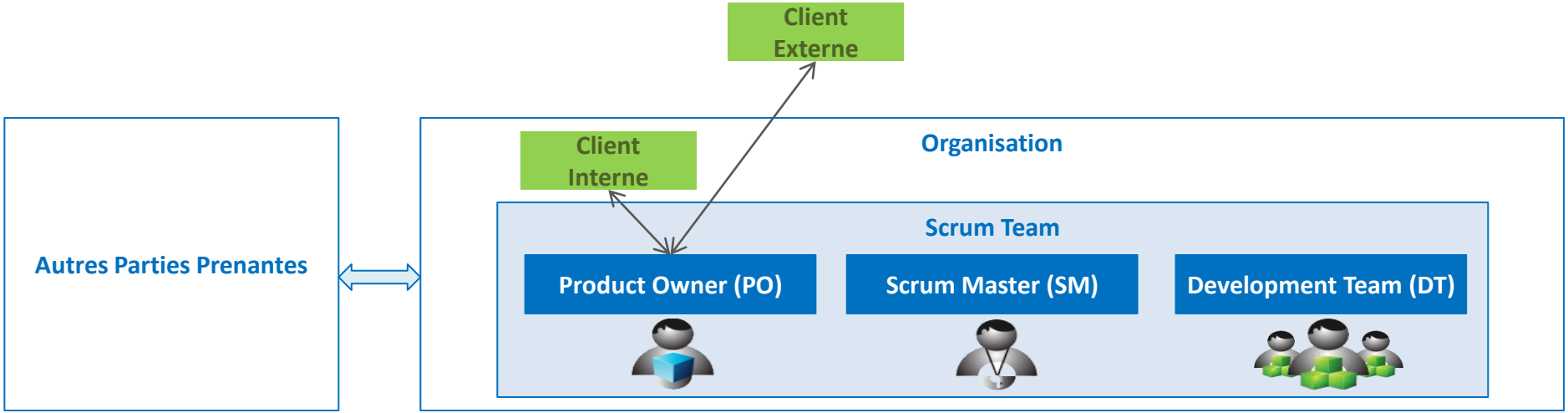
1. *Aperçu de l'équipe*
2. *Product Owner*
3. *Scrum Master*
4. *Development Team*
5. *Autres rôles*

- **Incrément 3 – Et après**
  - Sprint 1 Appliquer Scrum dans la réalité
  - Sprint 2 Ressources & Certifications

The SCRUM Team 1/5

Aperçu de l'équipe : Il n'y a (que) 3 types de roles dans un Projet Scrum

→ Une même personne peut occuper plusieurs roles, mais ce n'est pas recommandé



1 personne
Temps plein / temps partiel
Orienté métier/fonctionnel
Interface avec le client

1 personne
Temps plein / temps partiel
Coach Scrum et facilitateur, servant leader (au service des autres)

3 à 9 personnes
Temps plein (autant que possible)
Spécialistes, experts
→ Ceux qui produisent

The SCRUM Team 2/5

Product Owner *(une personne)*

- C'est une personne de l'organisation, qui est **orientée business.** Son but est de ***maximaliser la valeur du produit*** et le travail de l'équipe de développement.

- Un Product Owner n'a pas besoin d'avoir une connaissance poussée (technique) dans le domaine du projet
  - → Dans le développement de logiciel, un Product Owner n'a pas besoin d'être un développeur lui même. En revanche il doit parfaitement maitriser le sujet fonctionnel du projet.

- Derrière le Product Owner il peut y avoir un comité, mais seule une personne doit endosser la responsabilité de ce rôle afin de garantir la prise de décisions.

- Il/elle est responsable des points suivants :
  - → ***Créer et raffiner le Product Backlog.*** C'est un artefact central qui consiste en une liste priorisée d'items (user stories) que le client attend sur le projet.
  - → ***Définir des User Stories claires et compréhensibles*** par l'équipe de développement, et les autres parties prenantes.
  - → ***Communiquer efficacement*** avec le client, et faire en sorte de maintenir à jour son Product Backlog en prenant en compte tous les changements.
  - → ***Mesurer la performance*** du projet, ***faire du prévisionnel***, et ***rendre cette information transparente*** à toutes les parties prenantes.

The SCRUM Team 3/5

Scrum Master *(une personne)*

- C'est une personne qui comprend et maitrise Scrum, dont le but est d'aider l'équipe Scrum en la coachant, et en garantissant qu'elle suit et implémente correctement les processus Scrum.

- Scrum considère ce rôle comme une ***position de management***, qui gère les processus scrum plutôt que l'équipe Scrum.
  → Il/elle est un "servant-leader" au service de l'équipe Scrum.

- Les responsabilités du Scrum Master sont les suivantes :
  - ***Essayer de résoudre les obstacles rencontrés*** par l'équipe de développement, ***faciliter*** les cérémonies Scrum, ***former*** et ***coacher*** l'équipe.
  - ***Aider*** ou conseiller le Product Owner dans la mise en place de processus et dans la communication de l'information.
  - ***Aider les parties prenantes extérieures à comprendre et agir*** correctement avec l'équipe Scrum dans le but de ***maximaliser la valeur*** créée par celle-ci.

- Le Scrum Master aide et pilote la phase de transition d'une organisation vers la méthodologie Scrum.

- Etre un/une Scrum Master sur un projet **n'occupe pas toujours une personne à 100%**
  → Dans ce cas, Scrum recommande que la même personne soit Scrum Master sur plusieurs projets plutôt que d'en faire un développeur à temps partiel.

The SCRUM Team 4/5

Development Team *(équipe de 3 à 9 personnes)*

- Les membres de l'équipe de développement sont **des experts de différents domaines** qui sont responsables **ensemble** *:*
  - De la ***production(delivery)*** des éléments du Product Backlog,
  - De la ***supervision(management)*** de leur propre efforts.
  - → Les membres de l'équipe sont affectés à temps plein, afin de resté concentré et agile.
  - → La composition de l'équipe doit varier au minimum, afin de limiter les pertes de productivité.

- Les caractéristiques de l'équipe sont :
  - **cross-fonctionnelle**; être capable de faire le travail de A à Z de chacun des éléments du Product/Sprint Backlog.
  - **auto-organisée**; être en mesure de trouver sa propre manière de réaliser les tâches plutôt que de suivre des directives.
  - **alignée avec le but du projet** plutôt que de travailler en aveugle.

- Une tâche peut être affectée à un membre tout au long d'un Sprint, mais l'équipe entière est responsable de sa bonne implémentation
  - → Les tâches sont impersonnelles.

- L'équipe délivre le produit final, incrément par incrément tel que planifié avec le Product Owner. Elle travaille dans un mode de pensée orienté produit.

The SCRUM Team 5/5

Quel rôle pour les autres parties prenantes ?

Rôle du Manager

- **Faciliter le travail de l'équipe Scrum** :
  - Communiquer les informations importantes et les guidelines au Product Owner pour garantir la valeur ajoutée du produit
  - Soutenir le Scrum Master dans l'implémentation du changement organisationnel
  - Soutenir l'équipe de développement et l'aider à s'auto-organiser
- **Respecter les processus Scrum** :
  - Pour soumettre des changements, le manager doit en discuter avec le Product Owner (pas de bypass)

Rôle du Client

- **Faciliter le travail de l'équipe Scrum** :
  - Communiquer les informations importantes et les guidelines au Product Owner pour garantir la valeur ajoutée du produit
- **Respecter les processus Scrum** :
  - Pour soumettre des changements, le client doit en discuter avec le Product Owner (pas de bypass)

Rôle du(es) Utilisateur(s)

- **Faciliter le travail de l'équipe Scrum** :
  - Communiquer les informations importantes et du feedback au Product Owner pour garantir la valeur ajoutée du produit

Increment 2 - Sprint 3

- **Incrément 1 – Gestion de projet agile**
  - Sprint 1 Pourquoi Agile ?
  - Sprint 2 Manifeste Agile, historique et points communs des méthodes
- **Incrément 2 – le Framework SCRUM**
  - Sprint 1 Introduction
  - Sprint 2 Equipe SCRUM
  - Sprint 3 Cérémonies SCRUM
  - Sprint 4 Artéfacts SCRUM
- **Incrément 3 – Et après**
  - Sprint 1 Appliquer Scrum dans la réalité
  - Sprint 2 Ressources & Certifications

1. *Release Planning*
2. *Les 4 rituels de Sprint (Ceremonials)*
   1. *Sprint Planning Meeting*
   2. *Sprint Daily Meeting*
   3. *Sprint Review Meeting*
   4. *Sprint Retrospective Meeting*

SCRUM Events 1/5

- **Etre Agile** ne veut pas dire **aucune planification (sur le long terme).**

→ Il est conseillé de produire un planning avec une vision moyen/long terme partagé avec la direction : le **release planning**

→ Re-planifier autant que nécessaire en incluant les changements soumis et validés (après chaque instance de Sprint)

***Répétition** du **même processus** encore et encore*

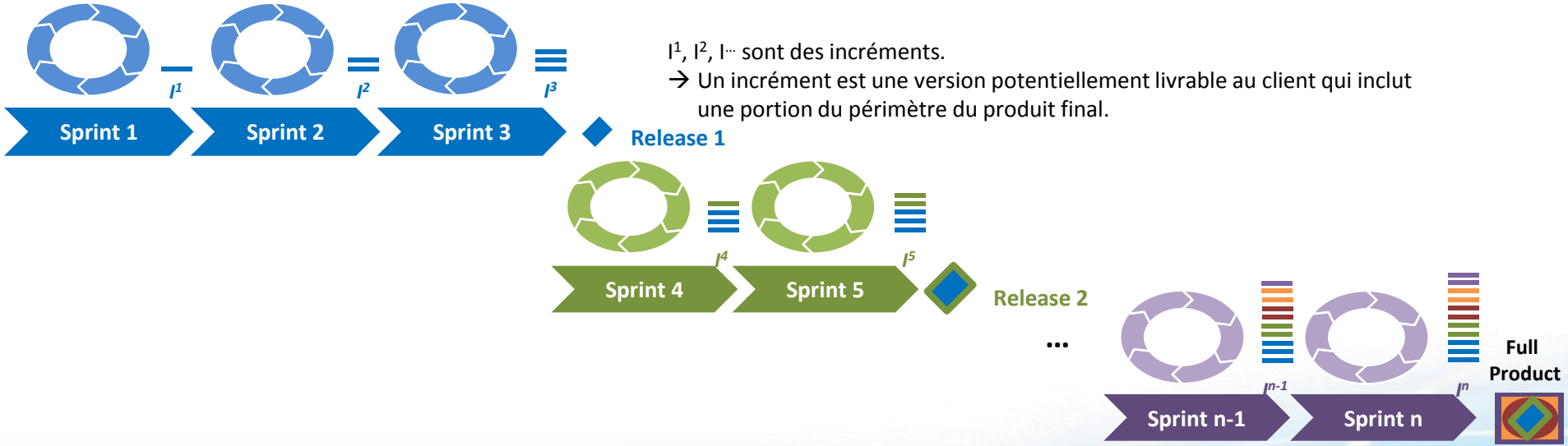
- Chaque projet Scrum délivre le produit final après un nombre de cycles ***itératifs*** et ***incrementaux***, appelés les **Sprints.**

→ Un Sprint est borné dans le temps, généralement de 1 à 4 semaines

→ La durée des Sprints doit demeurer constante

→ Un Sprint est un conteneur pour 4 événements Scrum, le développement d'un incrément et l'entretien du Product Backlog

*Production de **nouvelles fonctionnalités (valeur)** à chaque cycle*



I^1, I^2, $I^{\cdots}$ sont des incréments.

→ Un incrément est une version potentiellement livrable au client qui inclut une portion du périmètre du produit final.

SCRUM Events 2/5

Event #1 : Sprint Planning Meeting 1/2

- **Partie 1 : définir le QUOI ? (4h pour un Sprint de 4 semaines)**
  - Le Product Owner trie et ordonne les éléments du Product Backlog.
  - Le Product Owner vérifie que les stories soient compréhensibles et comprises de tous.
  - La Development Team estime la capacité de travail qu'elle peut délivrer dans le Sprint (en fonction de la vélocité).

  - Ensuite, l'équipe de développement sélectionne un nombre approprié d'éléments depuis le haut de la liste du Product Backlog, et les affecte au Sprint Backlog, afin de les réaliser dans le Sprint courant.

  → L'effort de travail de chaque item est estimée/confirmée par la Development Team

  → La somme de travail sélectionnée doit être proche de la capacité de travail de la Development Team.

  - A partir de la sélection des items, la Scrum Team va fixer un but au Sprint : le **Sprint Goal**.

  → Il fixe un objectif qui doit être atteint par le Sprint à travers l'implémentation des éléments du Sprint Backlog.

  → Il définit un cadre à la Development Team pour construire l'incrément.

Lorsque le **Sprint Backlog** et le **Sprint Goal** sont établis d'un commun accord avec le Product Owner,

→ Il faut ensuite établir **COMMENT** l'équipe va implémenter les user stories, de sorte à ce qu'elles soient "Done" et que le Sprint Goal soit atteint.

SCRUM Events 2/5

Event #1 : Sprint Planning Meeting 2/2

- **Partie 2 : définir le COMMENT ? (4h pour un Sprint de 4 semaines)**
  - Le **Sprint Planning** n'est pas nécessairement terminé au sein de cette réunion;
    - → Il "suffit" d'avoir un plan détaillé pour le début du Sprint,
    - → L'équipe peut préparer la suite du plan détaillé pendant le Sprint.

  - Chaque élément du Product Backlog est décomposé en un lot de **tâches** nécessaires afin de réaliser l'élément.
  - Chaque tâche doit être *estimée*, ordonnée en fonction de ses *dépendances*, et *caractérisée* avec ses propres critères d'acceptation.
  - Le Sprint Backlog est préparé par la Development Team, qui doit être en mesure de détailler et planifier la livraison des éléments du Sprint, et surtout comment l'équipe compte s'y prendre à travers.
  - Scrum n'impose aucune règle particulière pour la documentation, la décomposition en tâches et la présentation du Sprint Backlog.

Story	TO DO	...
Story 1	Task1, Task3, Task2	
Story 2	Task2, Task4, Task5, Task1, Task3	
Story 3	Task1, A compléter	

← Voici un exemple de Sprint Board.

Certaines tâches sont créées lors du *Sprint Planning meeting*, d'autres émergent au long du Sprint.

Les éléments du Sprint Backlog conservent généralement la même priorité que celle qu'ils avaient dans le Product Backlog, par conséquent, l'équipe doit d'abord travailler sur les éléments du haut.

SCRUM Events 3/5

Event #2 : Scrum Daily Meeting 1/2

- C'est une réunion quotidienne qui est **bornée dans le temps** à ***15 minutes***. Elle n'est pas nécessairement debout !

- L'heure et le lieu doivent demeurer constants
  - → plus simple et plus clair pour tout le monde.
  - → Pas d'excuses non plus!

- Cette réunion est dédiée à l'équipe de développement pour rester productive tout au long du Sprint.
  - → Ce n'est pas une réunion d'information pour les parties prenantes,
  - → L'intérêt premier est **l'inspection** et **l'adaptabilité** depuis la dernière réunion,
  - → Parallèlement, c'est l'opportunité pour être **transparent** avec les autres, et établir une **synchronisation** du travail à court terme.

- Pendant le Daily Scrum, chaque membre de l'équipe de développement répond aux 3 questions suivantes :

1. **Qu'ai-je accompli depuis la dernière réunion ?**
2. **Qu'ai-je prévu de faire avant la prochaine réunion ?**
3. **Quels sont les obstacles que je rencontre ?**

SCRUM Events 3/5

Event #2 : Scrum Daily Meeting 2/2

- Cette réunion est généralement l'opportunité de :
  - → **Evaluer la progression** par rapport au ***Sprint Goal.*** C'est usuellement une bonne idée d'afficher de manière visible et transparent le ***Sprint Board*** pendant le Daily Scrum.
  - → **Prévoir** la probabilité d'achever le Sprint comme planifié.

- Exemple de Sprint Board :

Sprint Goal : Gérer les fonctionnalités liées à l'enregistrement d'un utilisateur dans le système.

Story	TO DO	IN PROGRESS	TO VERIFY	DONE
Story 1	Task7 Task8 Task7	Task4 Task6	Task2 Task5 Task3	Task1
Story 2	Task9 Task4 Task10 Task8 Task7	Task3 Task6 Task5	Task1	Task2
Story 3	Task4 Task5	Task3 Task1	Task2	
Story 4	Task3 Task5 Task6 Task4	Task1		

SCRUM Events 4/5

Event #3 : Sprint Review Meeting

- L'objectif du Sprint est de produire un nouvel incrément.
  - → Par conséquent, à la fin du Sprint, il convient de vérifier que l'incrément produit est utilisable et de le valider avec le PO.

- C'est l'objet de la **Sprint Review Meeting**. **(4h pour un Sprint de 4 semaines)**
  - → L'équipe effectue une restitution de ce qui a été élaboré dans le Sprint
  - → C'est en général une **démo** organisée des nouvelles fonctionnalités
  - → La réunion est conçue pour **inspecter** l'incrément et collecter du feedback qui aide à améliorer ou à mettre à jour la description des éléments du Product Backlog.

- En général, cette réunion est **intentionnellement assez informelle**, (pas de PowerPoint et peu de préparation)
  - → Ce rituel ne doit pas devenir une distraction ni être esquivé par l'équipe; c'est la prolongation naturelle du développement.

- Parmi les participants on retrouve :
  - → Les 3 rôles clés : le *Product Owner*, la *Development Team*, le *Scrum Master*,
  - → Et, potentiellement, le management, le client, des intervenants extérieurs si besoin (interfaces avec d'autres projets, utilisateurs, …)

- C'est une réunion importante qui permet de valider le Sprint face au Sprint Goal et au planning réalisé en début de Sprint.
  - → **Dans l'idéal,** l'équipe a terminé et validé chaque user story, **MAIS** le plus important c'est qu'elle ait atteint le Sprint Goal fixé pour le Sprint.

SCRUM Events 5/5

Event #4 : Sprint Retrospective Meeting

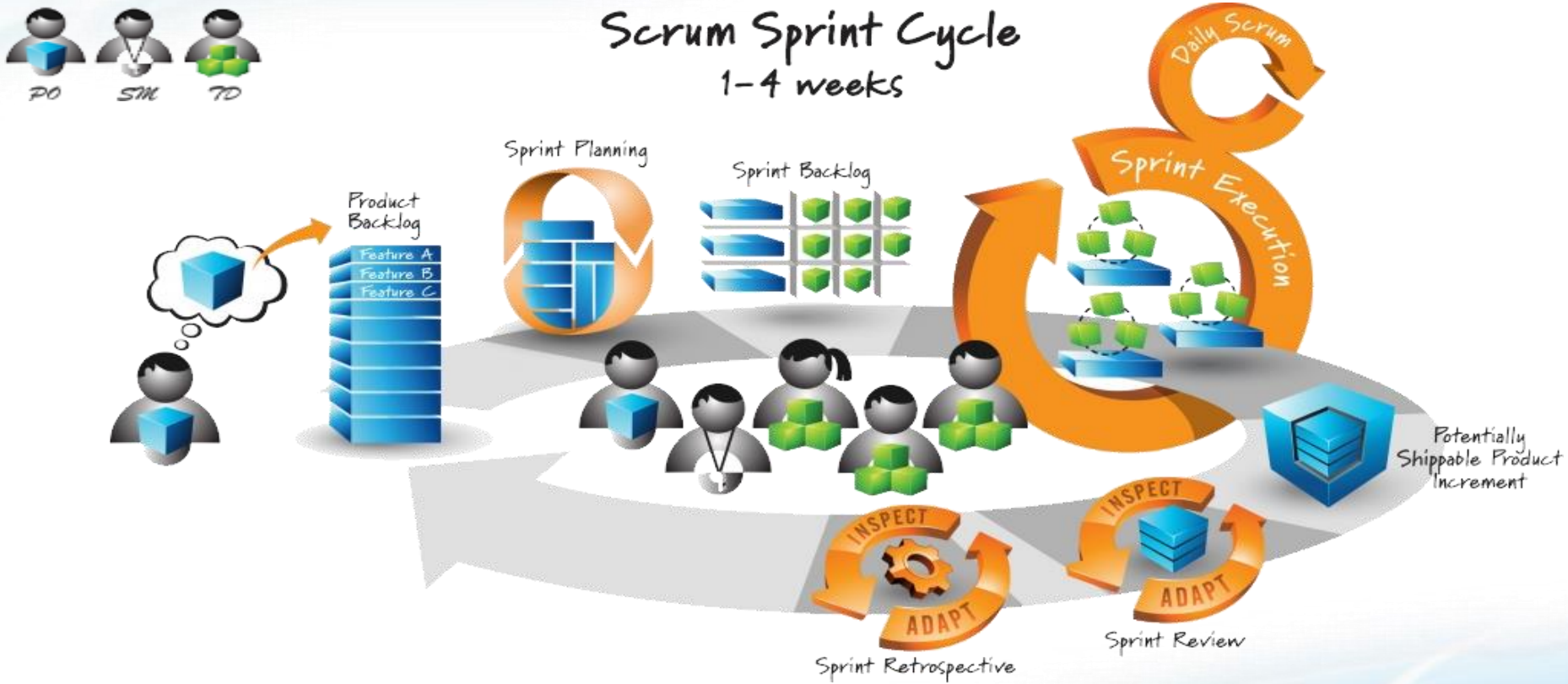
- **Le Sprint se termine par une dernière réunion : la Sprint Retrospective.**
  - → La plupart des équipes enchaine cette réunion juste après la *Sprint Review Meeting*,
  - → L'équipe entière assiste à cette réunion, ce qui inclus le *Scrum Master* et le *Product Owner*,
  - → C'est une réunion courte, qui dure en général moins d'une heure (Scrum propose une timebox de 3h pour un Sprint de 4 semaines).
  - → Parfois, un sujet piquant va alimenter les débats voire un conflit d'équipe, et dans ce cas l'escalade peut rallonger significativement la réunion.

- Dans tous les cas, peu importe la performance de l'équipe, **elle peut toujours s'améliorer** et capitaliser sur le Sprint réalisé.
  - → Une bonne équipe recherche en permanence des moyens pour s'améliorer.

- Une méthode *simple*, *efficace* et *recommandée* de réaliser cette réunion est d'utiliser la technique "start-stop-continue" : chaque membre répond aux 3 questions suivantes :
  - **Que doit-on commencer à faire ? (start doing)**
  - **Que doit-on arrêter de faire ? (Stop doing)**
  - **Que doit-on continuer à faire ? (Continue doing)**

- Le Scrum Master peut être amené à faciliter/arbitrer la réunion. Parmi les idées issues du brainstorming, l'équipe choisit d'un commun accord ou vote sur quelques (peu) axes d'améliorations à intégrer au prochain Sprint.

SCRUM Events (schéma récapitulatif)



Copyright (c) 2010, Innolution, LLC & Kenneth S. Rubin. All Rights Reserved.

Increment 4 - Sprint 4

- **Incrément 1 – Gestion de projet agile**
  - Sprint 1 Pourquoi Agile ?
  - Sprint 2 Manifeste Agile, historique et points communs des méthodes
- **Incrément 2 – le Framework SCRUM**
  - Sprint 1 Introduction
  - Sprint 2 Equipe SCRUM
  - Sprint 3 Cérémonies SCRUM
  - Sprint 4 Artéfacts SCRUM
- **Incrément 3 – Et après**
  - Sprint 1 Appliquer Scrum dans la réalité
  - Sprint 2 Ressources & Certifications

1. *Product Backlog*
2. *Definition of Done (DoD)*
3. *Definition of Ready (DoR)*
4. *Sprint Backlog*
5. *Incrément*

SCRUM Artifacts 1/4

Product Backlog 1/5

- Le **Product Backlog** (PB) est une liste priorisée des fonctionnalités attendues pour le produit.
  - → Le PO priorise les éléments comme il l'entend (MoSCoW, Valeur). L'effort et la complexité sont en revanche estimés par la DT.

- Il n'est PAS nécessaire de démarrer avec une liste exhaustive d'éléments dans le PB.
  - → En début de projet, le Product Owner identifie et transpose toutes les fonctionnalités sous forme d'user stories
  - → L'équipe de développement apporte sa contribution et contribue au *Backlog Grooming* (affinage, clarification, nettoyage, …)
  - → Ce premier Product Backlog est généralement suffisant pour démarrer le premier Sprint.

- Le Product Backlog est le miroir du traditionnel *Plan de Management*, sauf qu'il lui est permis d'évoluer et de changer au fur et à mesure qu'on en apprend plus sur le projet, le produit et ses utilisateurs.

- Un Product Backlog classique peut inclure différentes familles d'éléments :
  - **Fonctionnalités (Features)**
  - **Bugs** (dette technique, aussi peu que possible)
  - **Travail technique**
  - **Montée en compétence**
  - **Assurance qualité**

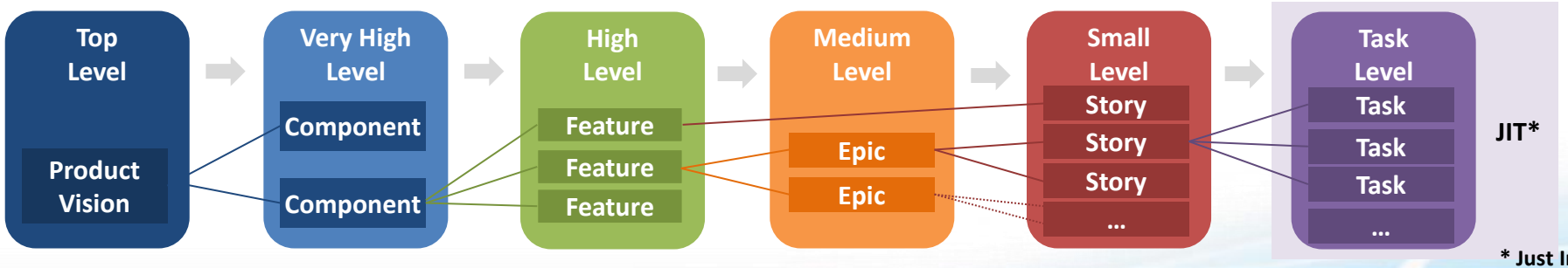
SCRUM Artifacts 1/4

Product Backlog 2/5

- Le Product Owner est **responsable** du Product Backlog, que ce soit sont ***contenu***, sa "***fraicheur"*** et son ***ordonnancement***.
- Un élément du Product Backlog (***PBI: Product Backlog Item***) est créé, caractérisé et raffiné avec l'équipe de développement.
- Les **User stories** sont un moyen efficace pour communiquer les besoins à satisfaire par le projet. Voici quelques exemples :
  - En tant que `<role>` je voudrais pouvoir faire `<action>` pour `<business value>`,
  - En tant que `<role utilisateur>` je peux `<story>` afin de `<bénéfice>`,
  - `<nom de personne>` veut `<story>` afin de `<bénéfice>`,
  - En tant que `<utilisateur>` je veux `<un but>` afin de `<objectif à atteindre>`.

← Scrum n'impose pas les User Story, mais recommande leur utilisation

- Les stories utilisateurs sont déclinées / décomposées depuis la vision du produit selon la hiérarchie suivante :



Product Backlog 3/5

- Pour gérer le Product Backlog, le Product Owner est invité à suivre la règle **DEEP** :

  - **Détaillé de manière appropriée.**
    - Les éléments prioritaires doivent être suffisamment bien compris de sorte à être correctement implémentés dans le sprint à venir (voir DoR).
    - Comme pour le *Rolling Wave Planning* en gestion classique, les éléments peu prioritaires, peuvent rester être grossièrement détaillés.

  - **Estimé *.**
    - Le Product Backlog n'est pas juste qu'une liste exhaustive du travail à effectuer → C'est également un outil de planification (à moduler)
    - * : les éléments du bas sont généralement moins bien compris et de ce fait, chiffré avec une fourchette.

  - **Emergent.**
    - Le Product Backlog est vivant. Il change et évolue avec le temps.
    - Les éléments du Product Backlog peuvent être ajoutés, supprimés, re-priorisés, …

  - **Priorisé.**
    - Le Product Backlog est trié avec l'élément le plus important en haut, et celui ayant le moins de valeur en bas.
    - Ainsi, l'équipe peut maximaliser la valeur du produit ou du système développé à chaque phase du projet.

SCRUM Artifacts 1/4

Product Backlog 4/5

Chaque élément du Product Backlog suit le cycle de vie suivant (plus ou moins vite) :

Emergent L'idée est soumise. Tout le monde peut soumettre des idées, mais seul le PO peut les approuver.

Approved Le PO a déterminer que la storie est en adéquation avec les objectifs du produit.
L'élément doit être caractérisé, analysé et priorisé.

Ready La story est suffisamment claire pour tout le monde avec des critères d'acceptation. (voir DoR)
La story est prête à être implémentée dans un Sprint.

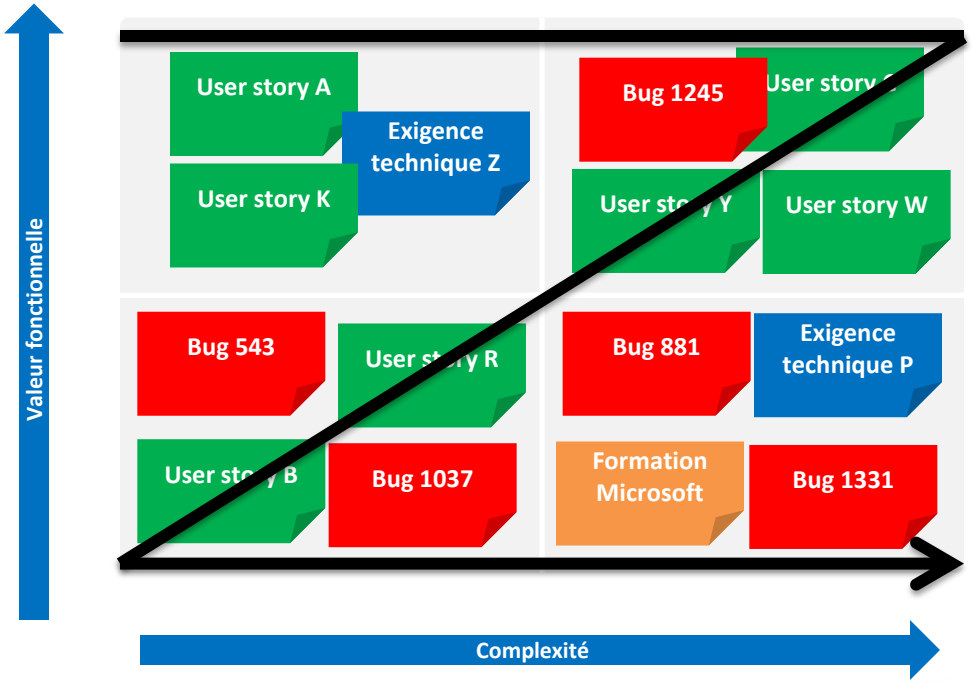
In progress Implémentation, cela peut prendre du temps …

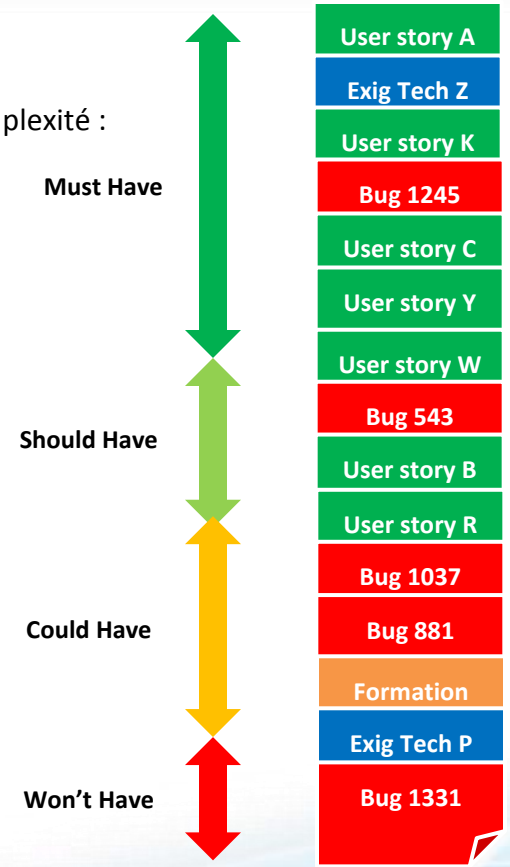
Done L'élément respecte tous les critères d'acceptation en accord avec sa ''**Definition of Done**''

SCRUM Artifacts 1/4

Product Backlog 5/5

- Voici schématiquement le Product Backlog obtenu après classification Valeur / Complexité :





Definition of Ready & Definition of Done 1/3

- **Definition of Done (DoD)**
  - Comme dans tout projet, il doit y avoir une **compréhension partagée** des critères d'acceptation et de la validation d'une story.
    - → C'est ce que l'on appelle la Definition of "Done". Ce sont des critères qui doivent être discutés et validés par l'équipe Scrum
    - → L'équipe rend ainsi **explicite** et **visible** les critères d'acceptation qu'une story ou qu'un incrément doit atteindre pour être validé et accepté. (Scrum s'appuie sur la méthode INVEST pour définir ces critères).
  - Il peut y avoir plusieurs DoD, à différents **niveaux d'acceptation**:
    - → Pour une **tâche** (ex: relecture de code, tests unitaires, couverture de code, …)
    - → Pour un **élément du Product Backlog** (ex: critères qualité, métriques, tests et la documentation nécessaire)
    - → Pour un **Sprint** (ex: revue de démonstration du système)
    - → Pour une **Release** (ex: note d'acceptation d'une release)
  - Lorsque plusieurs équipes Scrum travaillent sur un même projet, ces définitions ne sont pas toujours communes car la nature de leurs prestations peuvent varier :
    - → Dans ce cas, chaque équipe décrit ses propres définitions et produit ses incréments en accord avec leur DoD.
    - → Toutefois, l'intégration (la somme) de toutes ces Definition of "Done" doit garantir la faisabilité de la production de l'incrément.

SCRUM Artifacts 2/4

Definition of Ready & Definition of Done 2/3

- **L'acronyme INVEST**
  - Acronyme anglais qui est une aide de mémorisation de critères d'acceptation, c'est une checklist,
  - Il est utilisé pour évaluer la **qualité** d'une user story : si la story ne satisfait pas au moins l'un des critères, alors l'équipe voudra surement la
    - Compléter ou reformuler,
    - Ou carrément la ré-écrire depuis une feuille vierge.

- **Les principes de la méthode INVEST** :
  - **I** mmediatly actionnable : Indépendante & libre de tout blocage externe (pas de dépendance forte)
  - **N** egotiable : Suffisamment détaillée pour apporter un support de débat et de discussion
  - **V** aluable : Porte une valeur clairement partagée par les parties prenantes
  - **E** stimable : Assez claire pour que l'équipe puisse l'estimer facilement [et précisément]
  - **S** mall : Divisée suffisamment en petits blocs afin d'être réalisée dans le Sprint
  - **T** estable : Contient des critères d'acceptation précis pour savoir "*quand la story sera validée*"

SCRUM Artifacts 2/4

Definition of Ready & Definition of Done 3/3

- **Definition of Ready (DoR)**
  - → Par analogie avec la "***Definition of Done***", l'équipe rend **explicite** et **visible** les critères (également basés sur la méthode INVEST)
  - → qu'une user story **doit satisfaire** afin d'être considérée comme **prête** et **mature**

- **Bénéfices**
  - Evite de commencer sur une activité que l'on a pas encore clairement définie,
  - → Ce qui se traduit par **des allers-retours** et des **débats coûteux** et/ou des **reworks**
  - Permet de filtrer ou de remettre à plus tard les activités mal-définies

SCRUM Artifacts 3/4

Sprint Backlog 1/2

- RAPPELS : Le **Sprint Backlog** est une liste de **tâches** identifiées par l'équipe pour mener à bien le Sprint.
  - Pendant la réunion *Sprint Planning Meeting*, l'équipe sélectionne des stories parmi celles qui sont prioritaires (*Voir Sprint Planning Meeting*),
  - Ensuite, l'équipe décompose chaque story en un lot de tâches pendant la réunion ou après,
  - La plupart des équipes terminent en estimant en nombre d'heures le temps alloué à chaque tâche afin de la mener à bien (de bout en bout).

- **L'équipe de développement**
  - Est **responsable** du Sprint Backlog : elle le gère par elle-même et est s'engage en termes de résultats,
  - Sélectionne les éléments du Product Backlog en accord avec le PO, et en fonction de sa *capacité de production* et de la *durée du Sprint*.
  - S'engage à réaliser les tâches qu'ils ont choisi → ce doit donc être les mêmes personnes.

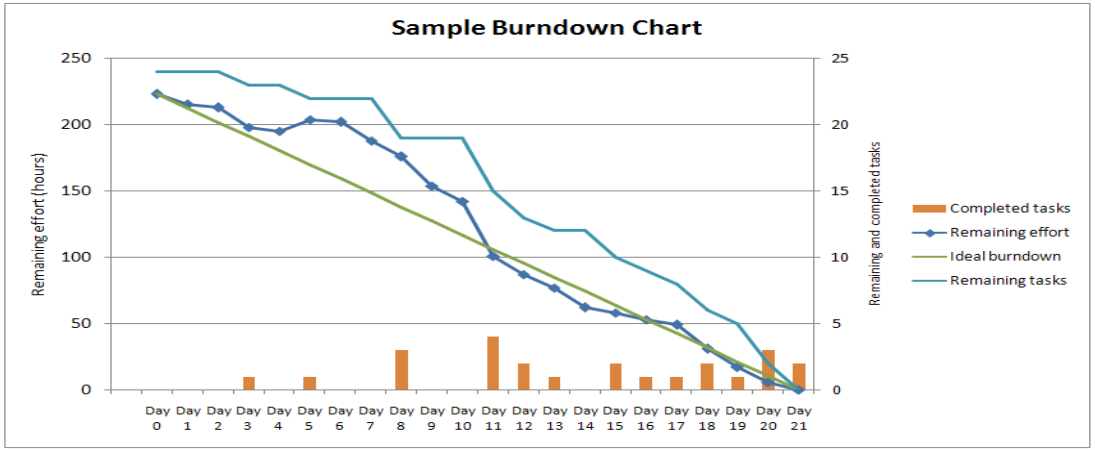
Sample Sprint Backlog

User Story	Tâche	Jour 1	Jour 2	Jour 3	Jour 4	Jour 5	Jour 6	Jour 7	Jour 8	Jour 9	Jour 10
En tant qu'administrateur je souhaite pouvoir gérer les utilisateurs de mon portail.	Concevoir le module utilisateur	4	0	0	0						
	Développer le module utilisateur	10	8	12	6						
	Rencontrer le spécialiste des utilisateurs …	2	2	0	0						
	Tester le module utilisateur	8	8	8	6						
En tant qu'administrateur je souhaite pouvoir gérer les enquêtes de satisfaction.	Ecrire le draft de l'enquête et le faire valider	12	16	14	11						
	Concevoir le questionnaire dans un logiciel de PAO	16	10	5	0						
	…										

SCRUM Artifacts 3/4

Sprint Backlog 2/2

- **Pendant le Sprint** :
  - Les membres de l'équipe sont *responsables de la mise à jour* du Sprint Backlog *au fur et à mesure des informations disponibles* (au moins 1 fois par jour).
  - Une fois par jour, l'estimation du travail restant est calculatée et restituée sous la forme d'un graphique par le Scrum Master → **sprint burndown chart**



- Dans certains Sprint **trop ou pas assez de travail** est défini pendant le *Sprint Planning Meeting*.

  → Dans ce cas, l'équipe peut ajouter ou supprimer du travail en accord avec le PO

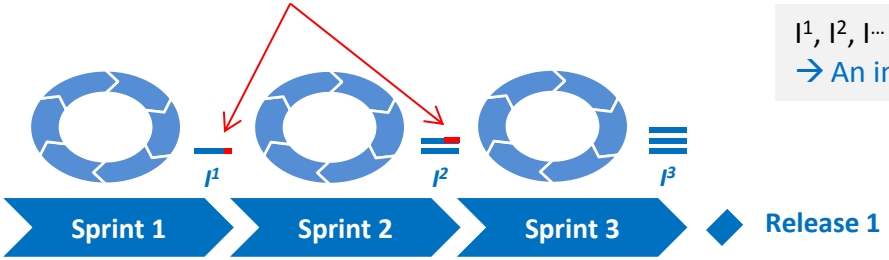
SCRUM Artifacts 4/4

Incrément

- La notion d'incrément de produit est une notion fondamentale à Scrum
  → C'est une portion du produit final fonctionnel qui **crée de la transparence** sur l'avancement du projet pour toutes les parties prenantes.

- Un Incrément est la **somme** de tous les éléments du Product Backlog complétés jusqu'à présent sur le projet, qui grossit continuellement
  → Chaque Incrément doit (devrait) apporter de nouvelles fonctionnalités (valeur) bien qu'il doit également considérer la dette technique
  → Chaque incrément doit être aligné avec la stratégie de l'équipe Scrum en terme de "*DoD*" et doit être validé par le Product Owner.

- La **Dette technique** représente l'accumulation d'un large backlog de faits techniques qui devront être résolus dans l'avenir
  → Ex : anomalies reliées au code, à l'environnement de développement, aux plateformes, COTS, conception, couverture de code, tests unitaires …
  → Les éléments de dette technique doivent être visibles, priorisés et traités de manière incrémentale de sorte à éviter un effet de masse plus tard



I^1, I^2, $I^{\cdots}$ sont des incréments.
→ An increment is a potentially releasable part of the final product.

Chaque incrément ***peut*** *(mais ne le sera pas forcément)* être ***livré***
→ Mais il ***devrait*** toujours être livrable : c'est le **PO qui décide**

EN général, le client émet du feedback et des requêtes de changement après réception de l'incrément ou pendant la *Sprint Review Meeting*,
→ Ces nouvelles requêtes entrent dans le Product Backlog

SCRUM Artifacts (Complément)

Story points

- Estimer n'est jamais trivial, en particulier lorsque le chiffrage doit être fait arbitrairement ou avec peu d'éléments.
  - → **L'estimation relative est un concept « simple »** :
    - tout le monde est d'accord sur le fait que préparer des pates est plus simple que de préparer un pot au feu
  - → **Par opposition, l'estimation absolue, en heures typiquement, est très subjective et volatile** :
    - la taille de l'équipe, l'expérience des membres de l'équipe ont une influence forte sur le chiffrage
    - 5h pour moi peuvent valoir 3h pour untel, et 10h pour un autre

- Pour chiffrer les éléments du backlog, Scrum recommande l'utilisation des **story points**, qui est une méthode d'estimation relative. Un story point permet d'évaluer l'effort qu'une équipe doit faire pour implémenter la story. On parle généralement de *taille* de la story. L'estimation en points prend en compte les critères suivants :
  - → L'effort de développement
  - → La complexité du développement
  - → Les risques et les incertitudes

- Pour un membre de l'équipe c'est généralement moins de pression : il y a moins d'engagement de résultat.
  - → A l'inverse, lorsque l'on dit ça va me prendre 10h, il faudra que ce soit terminé en 10h.

SCRUM Artifacts (Complément)

Story points

- Le principe est simple, encore faut-il définir des règles ou des story de référence. Scrum laisse à l'équipe le soin de choisir sa méthode relative :
  - Points linéaires : `1, 2, 3, 4`
  - Lettres : `A, B, C, D, E`
  - Tailles : `XS, S, M, L, XL, XXL`
  - Suite de Fibonacci : `1, 2, 3, 5, 8, 13, 20, 40 et 100`

On peut basculer facilement d'un mode absolu vers un mode relatif

L'inverse est beaucoup plus compliqué !

- **Processus d'estimation**
  1. Une User story est écrite et détaillée par le Product Owner
  2. Les membres de l'équipe discutent avec le PO, posent des questions pour avoir une vision claire et partagée du travail nécessaire
  3. Chaque membre de l'équipe estime la *taille* de la story. Il existe plusieurs techniques de chiffrage collectif
     a) La technique du Poker Planning (Suite de Fibonacci)
     b) La technique des doigts de la main
     c) Technique Delphi, WideBand, …
  4. L'équipe débat en fonction des résultats
  5. Le processus est itératif, et relancé tant qu'un consensus n'est pas atteint et partagé par toute l'équipe

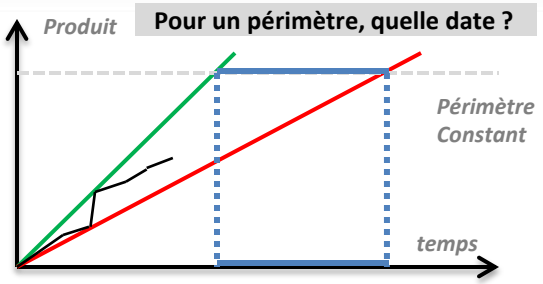
SCRUM Artifacts (Complément)

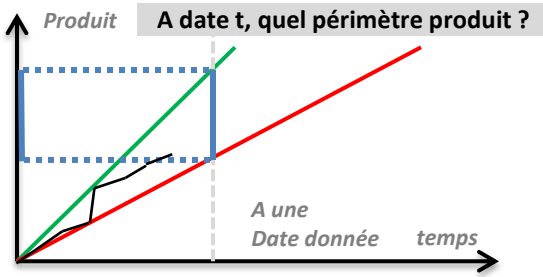
Vélocité

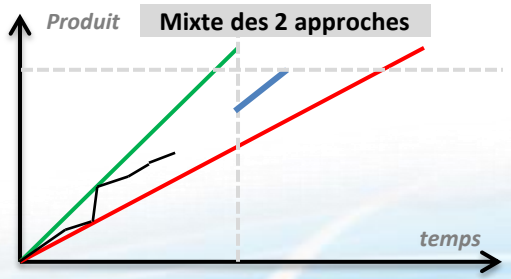
- Qu'est-ce que la **vélocité** ?
  - → C'est un moyen simple et efficace de mesurer la capacité de production d'une équipe en termes de fonctionnalités (story points).
  - → C'est une mesure sur du **constaté** qui peut servir à faire du forecast (approximation). On ne peut pas l'imposer !
  - → La vélocité est la somme de tous les story points des fonctionnalités validées (qui répondent à leur DoD).
  - → Le retour d'expérience de Scrum montre qu'en moyenne, la vélocité se stabilise au bout de 5 itérations
  - → On ne peut pas comparer la vélocité de 2 équipes différentes. Le changement des membres au sein d'une équipe remet souvent en cause la vélocité passée et donc la fiabilité du prévisionnel.
  - → On peut ainsi piloter 3 indicateurs pour estimer (de manière macroscopique)

- **Exemple:** soit une itération avec 4 user stories : A = 2 pts; B = 2pts; C = 3pts ; D = 1 pt.
  A la fin de l'itération, A, B et D sont terminées à 100% mais C n'est terminée qu'à 75%.
  → Vélocité de 5 pts

- **Exemple 2** : Le projet compte encore 30 pts, et la vélocité moyenne s'est stabilisée autour de 5. On peut approximer la date de fin du projet : entre 5 et 7 itérations.







Increment 3

- **Incrément 1 – Gestion de projet agile**
  - Sprint 1 Pourquoi Agile ?
  - Sprint 2 Manifeste Agile, historique et points communs des méthodes
- **Incrément 2 – le Framework SCRUM**
  - Sprint 1 Introduction
  - Sprint 2 Equipe SCRUM
  - Sprint 3 Cérémonies SCRUM
  - Sprint 4 Artéfacts SCRUM
- **Incrément 3 – Et après**
  - **Sprint 1** Appliquer Scrum dans la réalité
  - Sprint 2 Ressources & Certifications

Appliquer Scrum dans la vie réelle 1/2

- Scrum est une méthode facile à comprendre et à apprendre. Une organisation peut donc facilement se lancer dans ce mode Agile.
  → **Mais, Scrum est complexe et difficile à maitriser. Il y a un vrai gap entre la théorie et la pratique efficace.**

- Etre agile ne signifie pas être plus performant. L'atteinte de l'objectif est plus souvent atteint (vrai d'un point de vue client) mais souvent au détriment des coûts.

- Nombreuses sont les **difficultés** et les **caps** à franchir :
  1) Eduquer/former l'organisation et TOUTES les parties prenantes .
  2) Oublier le mode de pensée du fonctionnement traditionnel.
  3) Constituer une équipe auto-organisée est loin d'être simple, surtout pour des novices de la méthode.
     - Que faire des team leaders et chefs de projet qui avaient l'habitude de tout piloter.
     - Comment dispatcher, déléguer les responsabilités et les actions, et travailler avec le top management ?
     - L'équipe doit apprendre à travailler ensemble et à se discipliner.
  4) Avoir un seul et unique Product Owner est plus facile à dire qu'à faire ...
  5) Déterminer en début de projet (sans donnée passé) la limite de temps (time-box) d'un Sprint sans expérience.
  6) Apprendre à travailler en points plutôt qu'en jours-homme est un exercice compliqué et nécessite de bonnes références.
  7) Décomposer très finement les user-story est souvent chronophage et n'apporte pas de vraie valeur.
  8) Constituer un ensemble de processus pour garantir un minimum de suivi et un niveau de qualité satisfaisant.
  9) Etre constant et discipliné. Au début c'est facile, tout le monde est motivé, ensuite la contrainte opérationnelle ne plait pas à tous.
  10) Gérer le reste à faire ou le reste à produire n'est pas simple quand tout est géré en points. Les outils ne sont pas très précis.

Appliquer Scrum dans la vie réelle 2/2

- Adopter Scrum requiert un changement d'état d'esprit, d'accepter les défis, d'apprendre voire de ré-apprendre !

- Scrum ça se pratique, ça se vit avant d'avoir un point de vue sur la question :

  " To learn and not to do is really not to learn. To know and not to do is really not to know " **(S. R. Covey)**

- Apprendre et pratiquer le paradigme agile est très formateur. Certaines valeurs sont vraiment enrichissantes, et sans toutes les appliquer, certaines méritent d'être conservées dans l'application d'une méthode plus traditionnelle.
- Les daily meeting favorisent la communication mais ne doit pas être au détriment d'autres réunions
- La transparence pour l'avancement
- Les démos régulières pour collecter le feedback
- Le fait d'avancer à chaque fin de Sprint
- La boucle d'amélioration continue (as usual)
- Très enrichissant de se focaliser sur le produit par petits morceaux, mais compliqué à subdiviser parfois

- Partir de zéro (de la théorie) et l'appliquer sur le terrain c'est loin d'être évident
  - Fixer la time box du Sprint
  - Découpage et chiffrage des stories en story point
  - Une équipe n'est pas auto-organisée de nature
- Tout le monde n'apprécie pas forcément la fréquence des daily meeting et la discipline imposée (flicage)
- Décomposition en tâches parfois chronophage et inadapté
- Nécessite un gros effort de préparation et de suivi à la charge de l'équipe et du PO
- Le rework lié au fait que l'on a bâclé à la réunion de meeting
- La sensation de manque de contrôle pour un ancien PM
- Une gestion du RAF très compliquée

Mon REX personnel

Release PLAN

- **Incrément 1 – Gestion de projet agile**
  - Sprint 1 Pourquoi Agile ?
  - Sprint 2 Manifeste Agile, historique et points communs des méthodes
- **Incrément 2 – le Framework SCRUM**
  - Sprint 1 Introduction
  - Sprint 2 Equipe SCRUM
  - Sprint 3 Cérémonies SCRUM
  - Sprint 4 Artéfacts SCRUM
- **Incrément 3 – Et après**
  - Sprint 1 Appliquer Scrum dans la réalité
  - **Sprint 2** Ressources & Certifications

Resources & Certifications

- **Ressources SCRUM**

  - **Informations / whitepapers**
    - http://www.scrumguides.org/ (Scrum Guide Officiel)
    - http://www.infoq.com/minibooks/scrum-checklists (Scrum Checklists Libres)
    - https://www.scrum.org/Assessments/Open-Assessments (Tests d'évaluation libre pour le passage du PSM/PSD et PSPO)

  - **Outils Scrum en ligne :**
    - https://www.icescrum.com/
    - http://www.acunote.com/ (version payante uniquement)
    - http://www.scrumdesk.com (version payante uniquement)

  - **Resources pour préparer une certification ou améliorer ses connaissances :**
    - http://www.scrum.org (Certifications PSM / PSD / PSPO)
    - http://www.scrumalliance.org
    - http://mplaza.pm/product/scrum-master-training-manual/
    - http://www.scrumstudy.org (Certification Scrum Gratuite & Scrum Body of Knowledge)

Resources & Certifications

- **Certifications SCRUM pour aller plus loin…**

- **Scrum Alliance**
  - **CSM** *Certified Scrum Master*
  - **CSPO** *Certified Scrum Product Owner*
  - **CSD** *Certified Scrum Developer*
  - **CSP / CST** *Certified Scrum Professional / Trainer*

COURT	35 qst / 45'
ASSEZ FACILE	24/35 (69%)
CHER	400$ to 2500$
A RENOUVELER	Tous les 2 ans (100$)

- **Scrum.org**
  - **PSM** *Professional Scrum Master*
  - **PSPO** *Professional Scrum Product Owner*
  - **PSD** *Professional Scrum Developer*

PLUS LONG	80 qst / 60'
PLUS DUR	68/80 (85%)
ACCESSIBLE	150$ - PSM 200$ - PSD/PSPO

- **Autres certifications Scrum**
  - Scrum Study : SDC → SMC/SPOC/AEC → ESMC, based on SBoK (Scrum Body of Knowledge)
  - EXIN Agile Scrum Foundation → EXIN Agile Scrum Master
  - …

Merci, si vous avez des question, critiques ou suggestions, n’hésitez pas !

guillaume.bladier@gmail.com